



Faculdade de Tecnologia de Americana

Curso Superior de Tecnologia em Análise de Sistemas e Tecnologia

da Informação - Jogos Digitais

GAME LEVEL DESIGN UTILIZANDO UDK

FELYPE POTESTINO COSTA





Faculdade de Tecnologia de Americana

Curso Superior de Tecnologia em Análise de Sistemas e Tecnologia

da Informação - Jogos Digitais

GAME LEVEL DESIGN UTILIZANDO UDK

FELYPE POTESTINO COSTA

felypepotestinocosta@gmail.com

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise de Sistemas e Tecnologia da Informação, sob a orientação do Prof. Me. Kleber de Oliveira Andrade.

Área: Jogos Digitais

Americana, SP 2013

BANCA EXAMINADORA

Prof. Me. Kleber de Oliveira Andrade (Orientador)

Prof. Willian Fuertes Batista

Prof. Msc. Fernando José Ignácio

AGRADECIMENTOS

Gostaria de agradecer à instituição de ensino FATEC (Faculdade de Tecnologia) pelo conhecimento adquirido em todos esses anos.

Agradeço ao meu orientador, Kleber de Oliveira Andrade, pela atenção prestada durante essa etapa e também aos professores da banca examinadora.

E por ultimo, mas não menos importante, agradeço à minha família e namorada por ter me apoiado durante o curso e fornecido suporte às minhas decisões.

DEDICATÓRIA

Dedico esse trabalho a todos que tenham interesse ao tema e gostam de ampliar seu horizonte de ideias pesquisando, aprendendo e sempre questionando.

"Uma vida sem reflexão não merece ser vivida" (Sócrates).

RESUMO

O texto contido nesse documento tem como objetivo conceituar *level design*, apresentar a ferramenta UDK e demonstrar a utilização de alguns dos conceitos de *game level design* através da *game engine* UDK. A metodologia consiste na execução de passos pré-estabelecidos por um guia de desenvolvimento de *level* e demonstração dos resultados dessa aplicação. O resultado obtido com esse conceito foi o desenvolvimento de um *level* completo contendo todos os passos apresentados resultando em uma boa jogabilidade, riqueza em detalhes e um bom direcionamento do jogador dentro do *level* quanto ao seu objetivo. Levando a conclusão da necessidade de aplicar tais técnicas dentro dos jogos e explicando como uma boa ferramenta poderá auxiliar nessa aplicação.

Palavras Chave: Level Design, UDK, Passos Iniciais.

ABSTRACT

The text contained in this document aims to conceptualize level design, present the UDK tool and demonstrate the use of some of the concepts of game level design through the UDK game engine. The methodology consists in the execution of steps pre-established by a level development guide and demonstration of the results of it. The result obtained with this concept was the development of a complete level containing all the steps and resulting in a good gameplay, rich details and good goal guidance to player within the level. Leading to the conclusion of the need to apply such techniques within the games and explaining how a good tool may assist with this application.

Keywords: Level Design, UDK, Initial Steps.

SUMÁRIO

1.	IN	ΓRΟΙ	DUÇÃO	.12
2.	GΑ	ME	LEVEL DESIGN	.14
2	2.1.	Lev	el Design	.14
2	2.2.	A cı	riação de um level design	.15
	2.2	2.1.	Coletando Referências	.15
	2.2	2.2.	Planejar o Level	.16
	2.2	2.3.	Tema e Consistência	.17
	2.2	2.4.	Estilo	.19
	2.2	2.5.	Level conta Historias	.20
	2.2	2.6.	Escala e Proporções	.22
	2.2	2.7.	Limite de Fronteiras	.23
	2.2	2.8.	Ambientes Interativos e Perigo	.24
	2.2	2.9.	Detalhes	.25
	2.2	2.10.	Luz e Sombra	.26
3.	UN	IREA	AL DEVELOPMENT KIT (UDK)	.28
;	3.1.	Visa	ão Geral	.28
;	3.2.	Des	senvolvimento Multi-Plataformas	.28
;	3.3.	Unr	realScript e Kismet	.30
;	3.4.	Elei	mentos da ferramenta	.31
;	3.5.	Cria	ação de Level	.32
;	3.6.	Pes	squisa de Conteúdo	.33
;	3.7.	Cor	npilando e Distribuindo	.34
4.	DE	SEN	IVOLVIMENTO PRÁTICO DE LEVEL DESIGN	.36
4	4.1.	Col	eta de referências	.36
4	4.2.	Plaı	nejar o Level	.37

	4.3.	Tema e consistência	37
	4.4.	Estilo	38
	4.5.	Historia	39
	4.6.	Escala e proporções	40
	4.7.	Limite de fronteiras	40
	4.8.	Ambientes interativos	41
	4.9.	Detalhes	41
	4.10.	Luz e sombras.	42
5.	CC	NSIDERAÇÕES FINAIS	44
6	RF	FERÊNCIAS BIBLIOGRÁFICAS	46

LISTA DE FIGURAS

Figura 1 - Imagem do Jogo Cail of Duty®: Modern Warrare 3 (Cail Of Duty, 2013)	.15
Figura 2 - Imagem do jogo Little Big Planet Karting (Little Big Planet, 2013)	.16
Figura 3 - Imagem do Livro Ultimate Game Level Design Guide (Ultimate Ga	me
Level Design Guide, 2011)	.17
Figura 4 - Imagem do jogo Deus EX (Square Enix, 2013)	.18
Figura 5 - Imagem do jogo Trine 2 (Trine2, 2013)	.18
Figura 6 - Imagem do jogo Trine 2 (Trine2, 2013)	.19
Figura 7 - Imagem do jogo The Witcher Wild Hunt (Thewitcher, 2013)	.19
Figura 8 - Imagem do jogo Fable 2 (Lion head, 2013).	.20
Figura 9 - Imagem do jogo Left 4 Dead 2	.21
Figura 10 - Imagem do jogo Battlefield 3	.21
Figura 11 - Imagem do jogo Battlefield 3	.22
Figura 12 - Imagem do Livro Ultimate Game Level Design Guide (Galuzin, 2011)	.23
Figura 13 - Imagem do jogo Left for Dead 2 (The Tech Report, 2013)	.23
Figura 14 - Imagem do jogo Minecraft (Empire Minecraft, 2013)	.24
Figura 15 - Imagem do jogo Crash Bandicoot 2: The Eel Deal (Crash Bandicoot	t 2:
The Eel Deal, 2013).	.25
Figura 16 - Imagem do jogo Tomb Raider (Tomb Raider, 2013)	.26
Figura 17 - Imagem do jogo Grand Theft Auto (Rock Star Games, 2013)	.26
Figura 18 - Imagem do emulador de iOS da UDK (Epic Games, 2013b)	.29
Figura 19 - Representação de programação feita no Kismet (Epic Games, 2013f).	.30
Figura 20 - Interface da UDK (Epic Games, 2013j)	.32
Figura 21 - Content Browser (Epic Games, 2013h)	.34
Figura 22 - Unreal Frontend (Epic Games, 2013i)	.35
Figura 23 - Imagem de vários desertos (Plantier, 2013)	.36
Figura 24 - Imagem do planejamento do deserto (do autor, 2013)	.37
Figura 25 - Imagem do reino dentro do level (do autor, 2013)	.38
Figura 26 - Imagem do estilo do level (do autor, 2013).	.39
Figura 27 - Imagem que conta historia do level (do autor, 2013)	.39
Figura 28 - Imagem demonstrando escala do level (do autor, 2013)	.40
Figura 29 - Imagem demonstrando limite de fronteiras (do autor, 2013)	.41

Figura 30 - Imagem demonstrando ambientes interativos (do autor, 2013)	.41
Figura 31 - Imagem demonstrando detalhes do level (do autor, 2013)	.42
Figura 32 - Imagem demonstrando Luz e Sombra (do autor, 2013)	.42

1. INTRODUÇÃO

Atualmente os jogos fazem parte do cotidiano, seja para entretenimento ou para fins profissionais como treinamento e simulação. Segundo Huizinga (2012, p. 5), jogos são:

"[...] O jogo é mais do que um fenômeno fisiológico ou um reflexo psicológico. Ultrapassa os limites da atividade puramente física ou biológica. É uma função significante, isto é, encerra um determinado sentido. No jogo existe alguma coisa "em jogo" que transcende as necessidades imediatas da vida e confere um sentido à ação. Todo jogo significa alguma coisa. Não se explica nada chamando "instinto" ao princípio ativo que constitui a essência do jogo; chamar-lhe "espírito" ou "vontade" seria dizer demasiado. Seja qual for a forma como o considerem, o simples fato de o jogo encerrar um sentido implica a presença de um elemento não material em sua própria essência."

Os jogos podem ter uma importância que vai além da compreensão lógica que é usada em nosso cotidiano. Huizinga (2012) mostra que existe um homo ludens que utiliza o lúdico¹ através de novas experiências. Essa influencia lúdica leva os jogos a uma crescente evolução, pois com a revolução tecnologica, eles passam a abranger o cotidiano das pessoas de forma mais direta sendo utilizando em vários segmentos do nosso mercado.

Segundo Orrico (2012), o mercado de jogos brasileiro é o quarto maior do mundo. Isso se tornou realidade devido a grande quantidade de brasileiros que gostam desde jogos casuais até superproduções das grandes empresas. Nesse

-

¹ Que faz referência a jogo ou brinquedos: brincadeiras lúdicas. Que tem o divertimento acima de qualquer outro propósito (Dicio, 2013).

ponto, não se pode ignorar o estudo necessário para fazer essas produções um sucesso e é quando tem início ao que se denomina *game design*. Segundo Tavares (2005), *game design* não pode ser definido apenas como a arte do jogo ou programação e sim como a ideia principal onde todo o universo do jogo deve ser moldado, para deixar o jogo atrativo e imersivo, com intenção de prender o jogador por horas em um mundo lúdico.

Existem vários elementos que compõe um documento de *game design* (GDD), um deles denomina-se *game level design*. Atualmente sabe-se que é improvável um jogo existir sem *level design*, pois pode-se dizer que o jogo é dividido em uma série de *levels* onde cada *level* pode ter como objetivo levar novas emoções e desafios ao jogador.

No início da tecnologia moderna os jogos eram divididos em níveis para facilitar o processamento, uma vez que os computadores não disponibilizavam muito desse recurso como atualmente. Porém, se realizar uma analise dos jogos atuais, até mesmo os jogos modernos continuam sendo divididos em níveis. Nota-se em alguns casos que essa divisão é dada de forma sutil para transmitir a impressão de um mundo aberto e livre (Paulin, 2013).

Então é notável a necessidade de se criar *levels* para jogos, sendo assim o objetivo central desse trabalho é demonstrar a utilização de alguns dos passos de *game level design* através da *game engine* ² *Unreal Development Kit* (UDK), evidenciando que o mesmo vai além de um simples trabalho artístico.

O restante deste trabalho está organizado em capítulos da seguinte forma: o segundo capítulo aborda conceitos de *game level design*, seus principais passos de aplicação e o objetivo de sua utilização dentro de um jogo; o terceiro capítulo apresenta as ferramentas utilizadas para desenvolver o *level*; o quarto capítulo demonstra a criação de um *level*; por fim a conclusão e os possíveis desenvolvimentos de trabalhos futuros são apresentados no capítulo cinco.

_

² *Game engine*, ou *engine*, é um programa de computador e/ou conjunto de bibliotecas, para simplificar e abstrair o desenvolvimento de jogos eletrônicos ou outras aplicações com gráficos em tempo real.

2. GAME LEVEL DESIGN

Este capítulo tratará da visão, segundo Galuzin (2011), sobre *game level* design discorrendo algumas das técnicas aplicadas na criação de cenarios e mostrando quais pontos um profissional responsável por este trabalho deve se atentar no dia-a-dia.

2.1. Level Design

O *level* de um jogo é um conjunto de elementos que visam gerar interação com o jogador sob o intuito de causar sensações de vários tipos, aumentando a imersão dos jogadores. O *level* é uma peça fundamental na criação de um jogo, pois nele serão aplicadas todas as ideias decorridas no documento de *game design* (McGuire e Jenkins, 2009).

Segundo Feil e Scattergood (2005), um *level design* é a junção de vários objetos de forma a criar harmonia com a intenção de apresentar um cenário que retrata a realidade do enredo proposto. É importante ressaltar que *level designers* não criam os modelos, texturas, personagens ou até mesmo o código fonte de um jogo, eles organizam todos esses elementos da forma mais adequada.

Torna-se importante entender qual a linha de pensamento do profissional que irá desenvolver o *level design*. O profissional deve ter uma visão abrangente ao olhar o mundo a sua volta, pois é essa visão que torna suas referências visuais em um level. Portanto para esse profissional é indispensável a utilização de fotografias e bloco de notas para anotar ideias e fazer esboços de locais do cotidiano, arquitetura de prédios, ideias de livros e tudo o que pode inspirar sua imaginação, essa base de conhecimento futuramente será utilizada como uma semente para o seu desenvolvimento (Galuzin, 2011).

2.2. A criação de um level design

Para criação de um bom *level design* é importante seguir alguns passos para a estruturação do mesmo. Há uma sugestão de cinquenta e cinco passos por Galuzin (2011a), porém esse trabalho tem por foco apresentar os passos básicos que são: coleta de referências; planejar o *level*; tema e consistência; estilo; historia; escala e proporções; limite de fronteiras; ambientes interativos e perigos; detalhes; luz e sombras.

2.2.1. Coletando Referências

Para dar início à produção de um *level* é necessário coletar referências de várias fontes como arquitetura, filmes, músicas, arte, fotografia e até mesmo de outros jogos. A Figura 1 apresenta um *level* que reproduz características de uma cidade dos dias atuais onde detalhes, como a pichação, se contrastam com estilos arquitetônicos do século passado.



Figura 1 - Imagem do jogo Call of Duty®: Modern Warfare 3 (Call Of Duty, 2013).

Referências ao cotidiano podem ser visualizadas em ambientes lúdicos, porém de formas mais sutis. A Figura 2 apresenta um mundo de fantasias onde é possível notar várias referências a uma pista de corrida, mas ao olhar com atenção

percebe-se que o mundo é construído de objetos diferentes como pano, papelão, rolhas, tampinhas de refrigerante e etc, dando sempre importância ao tema definido no GDD.



Figura 2 - Imagem do jogo Little Big Planet Karting (Little Big Planet, 2013)

2.2.2. Planejar o Level

O planejamento de um cenário é o ponto crucial, pois é nesse momento que o profissional colocará no papel seus pensamentos na forma de um esboço. Esse planejamento aumenta as possibilidades do ambiente tornar-se um trabalho sólido (Galuzin, 2011).

A Figura 3 mostra o planejamento proposto por Galuzin (2011) e algumas etapas de produção onde é decidida a ordem no qual o trabalho será desenvolvido, quais modelos e texturas serão utilizados e quais preocupações serão tomadas no decorrer da criação antes de abrir o editor UDK e colocar as ideias em prática. Devese salientar as proporções de seus objetos durante o processo de planejamento.

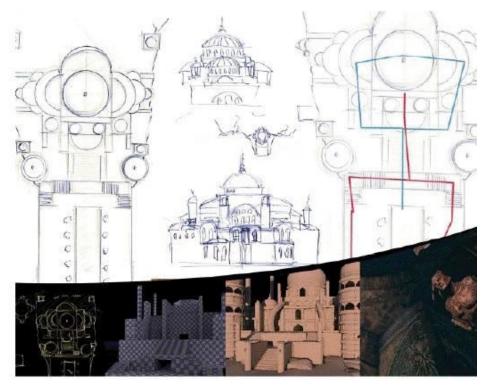


Figura 3 - Imagem do Livro Ultimate Game Level Design Guide (Ultimate Game Level Design Guide, 2011).

2.2.3. Tema e Consistência

O tema é a ideia principal de um *level* e seu objetivo é levar ao jogador o conceito de ambientação definido no GDD. Ele pode ser diferente dependendo da fase ou campanha que o jogador se encontra, porém ele deve-se manter consistente. Essa consistência pode ser medida pelo *game level designer* no momento da produção do cenário, verificando se sua visão não perder o foco principal (Galuzin, 2011).



Figura 4 - Imagem do jogo Deus EX (Square Enix, 2013).

Ao observar a Figura 4, pode-se ver que há grande menção ao tema futurístico proposto. No decorrer do jogo pode se notar claramente que o tema mantem-se consistente.



Figura 5 - Imagem do jogo Trine 2 (Trine2, 2013).



Figura 6 - Imagem do jogo Trine 2 (Trine2, 2013).

Observa-se nas Figuras 5 e 6 que há dois temas diferentes, porém a consistência não é afetada pela disparidade dos temas. No estilo nota-se nas Figuras 5 e 6, diferenças de cor e iluminação presente nos jogos.

2.2.4. Estilo

O estilo é responsável pelo impacto que o ambiente causa ao jogador, pois o mesmo expressa o tema proposto. Estilo é tudo que pode ser visto em um ambiente: arquiteturas, folhagem, colinas, montanhas, posicionamento de luz e sombras. Resumindo, e tudo que deixa o tema mais atrativo (Galuzin, 2011).



Figura 7 - Imagem do jogo The Witcher Wild Hunt (Thewitcher, 2013).



Figura 8 - Imagem do jogo Fable 2 (Lion head, 2013).

Observando-se as Figuras 7 e 8, compreende-se que ambos os jogos tem um estilo próprio e diferente. Porém suas histórias e temas propostos são similares, onde os personagens estão em um mundo no qual guerreiros com poderes mágicos tem que lutar para salvar o "mundo". Então fica uma dúvida no ar: porque existe tanta diferença na aparência? A resposta é bem simples: o público alvo. No jogo The Witcher Wild Hunt o público alvo é composto por adultos, e por isto possui um tema mais obscuro contendo muito sangue e violência. Já no jogo Fable 2, o público alvo é composto por crianças e pré-adolescentes, contendo um tema mais fantasioso, onde muitos de seus inimigos são criaturas místicas. Nota-se que o estilo do jogo influencia diretamente na historia contada.

2.2.5. Level conta Histórias

Como de fatos os jogos trazem uma gama de eventos que fazem com que a mente do ser humano viaje em outras historias. Galuzin (2011) afirma que o *level* deve contar uma história de forma que o jogador entenda o enredo apenas observando seu estilo e tema.



Figura 9 - Imagem do jogo Left 4 Dead 2.

Na Figura 9 é possível definir, mesmo que superficialmente, o enredo. Elementos como panos jogados ao chão e mensagens escritas na parede próxima à mesa (onde se encontram declarações de despedidas como se quem à escreveu já estivesse para morrer) demonstram que esse quarto foi um abrigo em tempos difíceis. Mesmo não sabendo o propósito do abrigo, sabe-se que o jogo tem caráter de sobrevivência, pois profissional se empenhou em contar uma história através dos objetos. É importante ressaltar que Galuzin (2011) afirma que o cenário pode contar histórias secundárias mesmo que não tenha conexões com a principal.



Figura 10 - Imagem do jogo Battlefield 3.

Na Figura 10 é retratada uma mesa onde há equipamentos que compõem uma bomba, a qual o jogador deverá desarmar, porém existem detalhes colocados pelo *level designer* que leva a outro caminho. Ao utilizar a mira do rifle para aproximar a visão da placa de circuitos, visualiza-se o logo da empresa japonesa Konami junto à numeração "GX775" como observado na Figura 11.



Figura 11 - Imagem do jogo Battlefield 3.

Caso o jogador realize uma busca em qualquer ferramenta de pesquisa, irá encontrar uma placa de circuitos do fliperama Super Contra de 1988, que pertence a Konami (Cordell, 2012). Com isso, percebesse a importância de um *level* contar uma história mesmo que ela não esteja diretamente ligada ao enredo, causando uma sensação de envolvimento do jogador. Para conseguir esse feito, um bom *level designer* deve realizar um planejamento detalhado em seu *level* mantendo boas escalas e proporções.

2.2.6. Escala e Proporções

Um *level designer* deve atenta-se às escalas e proporções colocadas no ambiente, pois são peças fundamentais e de grande importância para causar o efeito de imersão. Se uma porta for muito pequena ou muito grande ou estiver fora do contexto apresentado no GDD, pode-se quebrar completamente ou parcialmente a imersão do jogador. Para que isso não aconteça, todo o *level* deve ser baseado na escala de seu personagem (Galuzin, 2011).

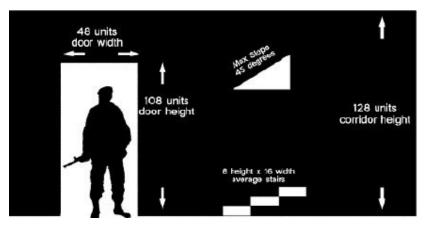


Figura 12 - Imagem do Livro Ultimate Game Level Design Guide (Galuzin, 2011).

Observa-se na Figura 12 que o tamanho do personagem influencia no tamanho de corredores, portas, escadas e etc, portanto é de vital importância que o *level designer* saiba como funcionam as unidades de medidas do editor escolhido para realizar o trabalho. O tamanho do personagem influência direta no tamanho do *level* onde será necessário colocar os limites de fronteiras.

2.2.7. Limite de Fronteiras

O limite de fronteiras é a parte que define até onde o jogador poderá ir. O fim de um *level*, segundo Galuzin (2011), é muito importante e deve deixar a impressão de que o *level* não termina, colocando objetos que fornecem a ideia de continuidade em uma área não jogável.

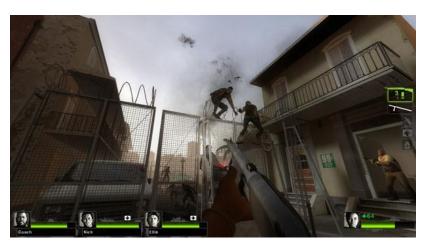


Figura 13 - Imagem do jogo Left for Dead 2 (The Tech Report, 2013).

A Figura 13 apresenta um *level* de um famoso jogo de zumbis onde seu objetivo é sobreviver. Mesmo com olhos atentos, não consegue-se notar o limite de fronteira, porém é bem visivel: a cerca na frente do jogador é o limite máximo. É importante observar que o *level designer* teve o cuidado de colocar carros, prédios e monstros para nascer do lado oposto ao limite, causando assim a sensação que o mundo não tem fim e que a cidade ao fundo está infestada de zumbis. Em alguns casos os limites podem ser ambientes interativos.

2.2.8. Ambientes Interativos e Perigo

Ambientes interativos são objetos no qual o jogador pode interagir com o *level*. Essa interação pode vir a ser parte do enredo principal ou não ter peso algum para a história, podendo ser um simples barril que pode se quebrar ou uma arma jogada no chão (Galuzin, 2011).



Figura 14 - Imagem do jogo Minecraft (Empire Minecraft, 2013).

O jogo da Figura 14 é incomum se abordamos o termo *level design* porque seus *levels* são gerados aleatoriamente por scripts. Mas, ao se tratar de interação com cenários, Minecraft é construído inteiramente dessa maneira, tornando todos os

componentes do jogo interagíveis e permitindo a reconstrução pelo jogador como achar melhor.

Em muitos casos, os ambientes interativos podem deixar aberturas para possiveis perigos de morte do personagem, como fogo atrás de uma porta fechada, armadilhas mortais, entre outros. O *level designer* deve atentar-se em passar a sensação de que algo está errado (Galuzin, 2011).



Figura 15 - Imagem do jogo Crash Bandicoot 2: The Eel Deal (Crash Bandicoot 2: The Eel Deal, 2013).

Na Figura 15 pode-se ver claramente que a água na frente do personagem está amarela. Durante o jogo além da cor, observa-se som de eletricidade, e insinua ao jogador que o personagem tomará um choque ao entrar em contato com a água. A sensação de perigo é colocada pelo próprio *level designer* com intuito de alertar o jogador. Estas técnicas alerta ao jogador têm por objetivo voltar sua atenção para os detalhes.

2.2.9. Detalhes

Os detalhes são importantes para dar vida ao cenário, porém não é necessário que o profissional coloque detalhes em todo o ambiente. Galuzin (2011) diz que deve-se colocar detalhes em que o jogador irá passar, como: Lixo, pedaços de construções, neblina, entre outros como demonstrado na Figura 16. Com isso, seu *level* causará mais imersão e terá traços mais bonitos que chamará atenção do jogador.



Figura 16 - Imagem do jogo Tomb Raider (Tomb Raider, 2013).

2.2.10. Luz e Sombra

Luz e sombra são recursos que dão a sensação de realidade ao seu *level*. Quando se fala em construção de *levels* é importante que o *level designer* não deixe a luz do seu ambiente uniforme. Ele deve gerar variações entre luz, causando diferentes tipos de sensações nos jogadores (Galuzin, 2011).



Figura 17 - Imagem do jogo Grand Theft Auto (Rock Star Games, 2013).

Observando a Figura 17, observa-se uma grande intensidade de luz retratando um dia de sol. Ao analisar a luz no *level*, nota-se que ela não é uniforme e isso torna-se visível quando olhamos os objetos brancos refletindo mais luminosidade. Todos esses detalhes de iluminação e sombras são colocados pelo *level designer*, em alguns casos a engine faz as sombras dos objetos em cena porém às vezes pode não agradar os olhos do profissional tendo então que refazer algumas ou todas as sombras dependendo da impressão que deseja-se passar ao jogador.

Como citado no início desse capitulo, esses passos foram apenas alguns dos cinquenta e cinco propostos para *level design* segundo Galuzin (2011). No próximo capítulo será apresentada a game engine UDK que é uma das muitas ferramentas presentes no desenvolvimento de um *level*.

3. UNREAL DEVELOPMENT KIT (UDK)

Nesse capítulo será apresentada uma visão geral da game engine UDK que se trata de um kit de desenvolvimento gratuito no mesmo molde da Unreal Engine 3 desenvolvido pela empresa Epic Games e que, segundo Lewis e Jacobson (2002) em uma análise de engines, é avaliada como uma ferramenta completa que facilita o desenvolvimento de aplicação para programadores sem muita experiência mesmo com ressalva de que a mesma requeria um computador robusto para a época em que foi lançada.

3.1. Visão Geral

Apesar de facilitar o desenvolvimento para pessoas sem muita experiência em programação, a UDK ainda é uma ferramenta complexa. Devido a sua variedade de recursos e elementos que permitem a criação de jogos completos de todos os tipos desde jogos simples e casuais até superproduções com efeitos visuais impressionantes. A sua interface possui inumeros botões e atalhos para diversas funções que são utilizadas pelo *level designer* no desenvolvimento (Epic Games, 2013a).

Essas vastas opções fazem com que um desenvolvedor iniciante tenha dificuldades de encontrar, o que é necessário para o seu desenvolvimento, por isso a utiliza-se um guia nos passos iniciais o que é de grande imporatancia. Para tanto a Epic Games (2013a) fornece em seu site, as tarefas necessárias para o desenvolvimento, tais como guias e scripts prontos para serem utilizados na ferramenta.

3.2. Desenvolvimento Multi-Plataformas

Na UDK existem especificações para cada tipo de plataforma que será suportada. A ferramenta suporta praticamente todas as grandes plataformas disponíveis no mercado (Epic Games, 2013a).

Para o iOS, a ferramenta UDK conta com um sistema de emulador já incluso que permite o desenvolvedor realizar testes de games utilizando o emulador de componentes físicos dos aparelhos Apple³ (Epic Games, 2013b) esse emulador é apresentando na Figura 18. Para as demais plataformas móveis, a UDK tem algumas limitações que podem ser removidas com a compra da versão profissional a Ureal Engine.



Figura 18 - Imagem do emulador de iOS da UDK (Epic Games, 2013b).

O grande ponto forte da ferramenta está em sua aplicação para a plataforma PC. Nessa plataforma pode-se explorar completamente o potencial da ferramenta gerando mundos com alta qualidade de imagem e sombra, havendo posibilidades futuras de rapidas migrações entre as plataformas PC, console e mobile (Epic Games, 2013c).

3 Apple Inc. é uma empresa multinacional norte-americana que tem o objetivo de projetar e comercializar produtos eletrônicos, software de computador e computadores pessoais. Os produtos de hardware mais conhecidos da empresa incluem a linha de computadores Macintosh, o iPod, o iPhone e o iPad.

.

3.3. UnrealScript e Kismet

O UnrealScript é a linguagem de programação que pode ser utilizada na UDK, porém esta linguagem é uma adaptação da linguagem C++, que possui um paradigma orientado a objetos, com heranças múltiplas (Epic Games, 2013d).O UrealScript é uma linguagem voltada para programadores e pode assustar iniciantes e *level designers* que desejam utilizar a ferramenta. Por esse motivo a UDK conta com um recurso diferente e muito funcional que é denominado Kismet, este recurso trata-se de uma programação indireta, na qual não há visibilidade do código. São apresentados organogramas que através de ligações podem resultar em interação com ambientes ou até funções pré-desenvolvidas pelo programador (Epic Games, 2013f).

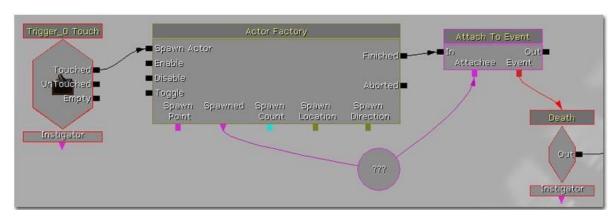


Figura 19 - Representação de programação feita no Kismet (Epic Games, 2013f).

A Figura 19 mostra um evento de morte sendo atribuído a um ator, esse ator pode ser o personagem principal, inimigos ou até mesmo elementos interativos. O elemento gatilho (*Trigger_0 Touch*) é um elemento que reage quando o jogador o toca, tendo retorno binário um para tocado e zero para não tocado. Há também a condição empty que verifica se existe algo dentro do elemento gatilho e retorna zero para falso e um para verdadeiro e, conectado a esse elemento, temos Fabrica de Atores (*Actor Factory*). Essa fabrica faz exatamente o que é descrito em seu nome ao dar início chamando a função *Spawn Actor*. Ela gera um ator e o armazena em

uma variável vazia representada com interrogações dentro de um circulo acima. Nesse mesmo momento, ao finalizar, ela chama o elemento Anexar ao Evento (*Attach To Event*) que compartilha a mesma variável já preenchida pela Fabrica de Atores. Com o ator capturado, ela executa o evento Morte (*Death*) que tem como alvo uma variável ator a qual o evento será disparado (Epic Games, 2013f).

3.4. Elementos da ferramenta

Na UDK pode-se encontrar objetos para construir um ambiente, para isso são disponibilizados elementos de estrutura básica de um jogo através das opções de tipo de jogo, visualização do jogo e interface (Epic Games, 2013a). O elemento tipo de jogo é responsável por criar as regras e normas pré establecidas pelo GDD e a implementação dessas regras pelo programador, irá controlar todas as ações através de *script*. Alguns exemplos seriam: gerar pontuação, determinar reinício de fases e fim de jogo. Existem tipos de jogos pré-determinados dentro da ferramenta que permitem ao *level designer* criar o seu mundo sem a necessidade de um programador, mas em casos de customizações será inevitável a criação de um novo tipo de jogo (Epic Games, 2013g).

O elemento visualização do jogo trata-se de um conjunto de sistema que controla a aparência e comportamento do jogador assim como a visão do mundo a sua volta através do controlador do jogador que converte os comandos em ações dentro do jogo (*PlayerControler*). Os comandos do controlador são carregados pelo peão para gerar a interação do jogador com animação, dano e outras ações físicas (*Pawn*), a câmera é responsável pela localização e rotação da visão do jogador sobre o jogo essa visão pode ser customizada e pode haver uma ou mais cameras (Epic Games, 2013g).

Interface é a disponibilização de informações na tela, durante o jogo os elementos que compõe uma interface na UDK são: *HUD, Canvas e Scaleform*. O heads up display (HUD) é uma classe responsável por exibir elementos sobrepostos na tela de jogo e instanciando os mesmo para cada jogador em sua janela de visualização. *Canvas* é utilizado para desenhar imagens e textos na tela conforme

requisitado pela HUD. Já a *Scaleform* é a integração da ferramenta *Adobe Flash Professional* com a UDK de forma a permitir a utilização de animações e facilitar a manipulação de elementos de interface executando comandos diretamente no *Flash* via *fscommand*⁴ (Epic Games, 2013g).

3.5. Criação de Level

A UDK trabalha muito bem com a criação de *levels*, podendo criar vários *levels*, além de *sub-levels* para facilitar o processamento. Esses *levels* podem ser carregados de forma a não gerar processamento desnecessário visando manter o jogo optimizado. Há também uma função para criação de *levels* denominada *Matinee*, responsável por criar animações de objetos, tornando possível criar *cinemáticas* e interações com o ambiente. A interface da UDK, conforme mostra a Figura 20, é clara e possui rápido acesso às suas ferramentas (Epic Games, 2013a).

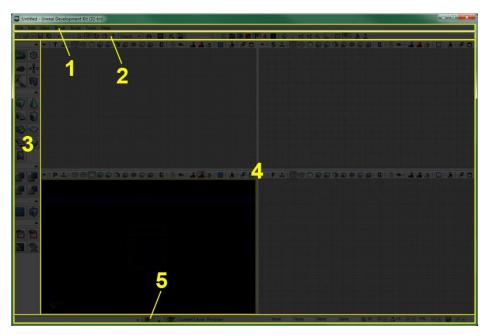


Figura 20 - Interface da UDK (Epic Games, 2013j).

4 Permite que o arquivo SWF se comunique com o Flash Player ou com o programa que hospeda o Flash Player, por exemplo, um navegador da Web. Você também pode usar a funçãofscommand() para transmitir mensagens para o Director ou para o Visual Basic, Visual C++ e outros programas que podem hospedar controles ActiveX.

- Barra de Menus no UDK é familiar para qualquer pessoa que tenha usado os aplicativos do Windows anteriormente. Ela fornece acesso a uma grande quantidade de ferramentas e comandos que são necessários para desenvolver seus levels (Epic Games, 2013j).
- 2) Barra de Ferramentas, como na maioria das aplicações, é um acesso rápido às ferramentas e comandos mais usados. Muitos dos itens encontrados nos menus da UDK também podem ser encontrados como botões na barra de ferramentas (Epic Games, 2013j).
- 3) Caixa de ferramentas é um conjunto de ferramentas utilizadas para desenvolver o level, podendo ajustar o brush para criar uma nova forma geometrica, e o controle de visibilidade e seleção de atores dentro das viewports (Epic Games, 2013j).
- 4) Viewports são as janelas para o mundo criado na UDK. Oferecendo múltiplas vistas ortográficas (superior, lateral, frontal) e uma vista com perspectiva, onde tem-se o controle total sobre o que é demonstrado em seu level (Epic Games, 2013j).
- 5) Barra de Status é responsável por alterar algumas opções e facilitar o desenvolvimento dentro da ferramenta. Nela pode-se encontrar a densidade da grade de edição e opções de configuração como salvar automaticamente (Epic Games, 2013j).

3.6. Pesquisa de Conteúdo

A UDK tem uma ferramenta chamada *content browser*, nesta ferramenta encontra-se uma gama de elementos prontos para serem utilizados, como sistemas de partículas, materiais, objetos 3D estáticos, esqueleto que permitem criar animações e texturas conforme demonstrado na Figura 21.

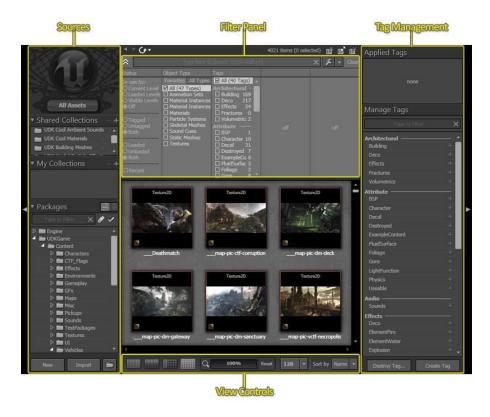


Figura 21 - Content Browser (Epic Games, 2013h).

3.7. Compilando e Distribuindo

A compilação muitas vezes pode ser complicada, porém a UDK tem uma ferramenta para ajudar esse processo que denomina-se *Unreal Frontend*, cujo o proposito é o de compilar a Figura 22 exibe a ferramenta Frontend que demonstra uma interface simples e permite que seja realizado teste em dispositivos móveis.

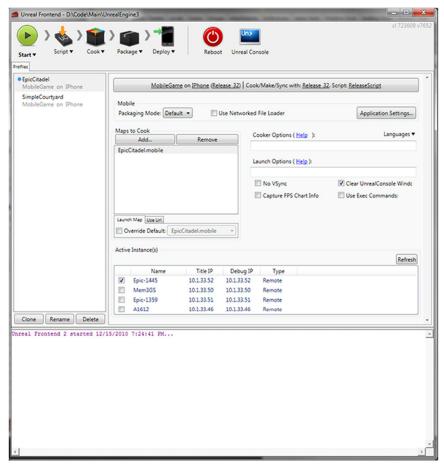


Figura 22 - Unreal Frontend (Epic Games, 2013i).

4. DESENVOLVIMENTO PRÁTICO DE LEVEL DESIGN

Nesse Capitulo será demonstrada a aplicação prática dos passos apresentados no capítulo dois com a intenção de gerar um cenário simples, bonito e imersivo. A proposta é gerar um ambiente desértico com pouca vegetação e a margem desse deserto existirá uma entrada para um reinado.

4.1. Coleta de referências

As referências coletadas para esse desenvolvimento foram de paisagens secas e de clima desértico conforme as diretrizes do GDD. A Figura 23 mostra vários desertos com diferentes detalhes e ângulos de visão. Essas imagens são importantes ao *level designers*, pois dará uma visão ampla de um ambiente árido e desértico demonstrando também variação de luz e sombras. Como foi demonstrado no capítulo dois, um *level designer* tem que utilizar diversas referências para gerar um trabalho completo.



Figura 23 - Imagem de vários desertos (Plantier, 2013).

4.2. Planejar o Level

Planejar o *level* é algo muito importante, este planejamento foram criados esboços para ajudar na escolha do posicionamento dos elementos na tela. Esses esboços serão seguidos fielmente, porém não existe uma obrigatoriedade para detalhes. O *level* pode ter um esboço bem simples desde que, através desse esboço, torne-se possível reproduzir o level com fidelidade.

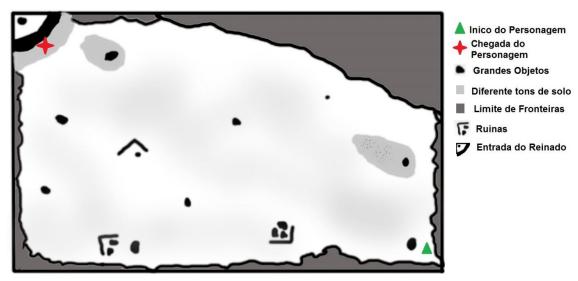


Figura 24 - Imagem do planejamento do deserto (do autor, 2013).

A simplicidade do esboço pode ser notada na Figura 24, onde os limites de fronteiras e o lugar onde o personagem deve chegar são demarcados. Os limites de fronteira são determinados pela parte cinza escura, as partes cinza claras denotam diferentes tipos de solo onde os solos tem alguma vida ou terão atrativos visuais interessantes, os pontos pretos espalhados são pedras de tamanhos variados e os traços negros retos são paredes de construções abandonadas. Os pontos verdes e vermelhos indicam respectivamente, o início e fim da jornada percorrida.

4.3. Tema e consistência

O tema e consistência do trabalho prático desenvolvido são dados por um deserto, porém foram adicionados elementos que aos poucos transformam em uma

entrada para um reino. A Figura 25 apresenta a existência de uma construção medieval, porém os tons de cores e os detalhes a sua volta passam a imagem de que esse é um reino que foi construído no deserto, por isso tons amarelos em suas paredes, grandes rochas e dunas a sua volta.



Figura 25 - Imagem do reino dentro do level (do autor, 2013).

4.4. Estilo

O Estilo adotado nesse cenário foi manter um deserto distribuindo os elementos de forma sutil e agradável, visando gerar uma sensação de curiosidade no jogador. Na Figura 26 pode-se notar ao fundo uma grande montanha de areia e mais a esquerda, na linha do horizonte, a entrada do reino. As diferentes tonalidades do solo unidos com arvores secas dão ao level um estilo que retrata uma zona desértica onde no passado houve vida e provavelmente florestas, levando o jogador a imaginar o porquê da existência desses elementos.



Figura 26 - Imagem do estilo do level (do autor, 2013).

4.5. Historia

Como foi demonstrado no capitulo dois, existe uma necessidade de que seja passada a historia do *level* e de seu jogo através dos elementos que irão compor o cenário para que o mesmo se mantenha com coesão. Se for observada a Figura 27, um robô está no meio de um deserto. Mas como poderia existir um robô em um deserto? Nesse ponto que se da atenção a historia contada pelos elementos do *level*, onde se notam naves que flutuam no ponto de início o que leva o jogador a imaginar uma possível invasão alienígena que tem como intuito explorar ou invadir o reino presente no deserto.



Figura 27 - Imagem que conta historia do level (do autor, 2013).

4.6. Escala e proporções

As escalas e proporções mantidas no *level* desenvolvido foram baseadas em torno do personagem principal, na qual suas proporções são aproximadamente as de um homem com um metro e oitenta centímetros de altura. Na Figura 28 nota-se que a porta principal de entrada possui um tamanho proporcional às grandezas reais. A entrada contem onze metros de altura por seis metros e trinta centímetros de largura, o suficiente para passar carruagens de cargas que poderiam ser utilizadas em tempos de reinado.



Figura 28 - Imagem demonstrando escala do level (do autor, 2013).

4.7. Limite de fronteiras

Os limites de fronteiras que foram utilizados no desenvolvimento constituem de montes de areia pedras e muralhas que impossibilitam o personagem de seguir a diante naquela direção. Na Figura 29 mostra-se um dos limites de fronteiras colocado no *level*. Esse limite de fronteira foi desenvolvido para ser natural, se encaixando perfeitamente no conceito de coesão do *level*. No reinado existem muralhas ao seu redor que podem impedir o jogador de prosseguir naquela direção, para lados opostos existem montanhas de pedra e areias que impossibilita o personagem de seguir adiante.



Figura 29 - Imagem demonstrando limite de fronteiras (do autor, 2013).

4.8. Ambientes interativos

Os ambientes interativos do *level* são elementos no qual o jogador pode interagir com o level. Para tanto, eles são colocados como parte da historia ou apenas como atrativo visual. A Figura 30 mostra o exemplo de ambiente interativo onde uma alavanca escondida pode ser acionada pelo jogador e, após esse acionamento, a porta do reino é aberta para que o personagem possa entrar, finalizando o *level*.



Figura 30 - Imagem demonstrando ambientes interativos (do autor, 2013).

4.9. Detalhes

Os detalhes em um *level* são partes importantes para gerar imersão e chamar atenção do jogador no cenário desenvolvido. Para esse ambiente foram escolhidos como elementos de imersão ruinas e oásis que enriquecem o nível de detalhes do *level*.



Figura 31 - Imagem demonstrando detalhes do level (do autor, 2013).

Na Figura 31, a presença de uma carroça com vários barris ao lado de um oásis são detalhes não necessariamente importantes para a história presente no *level*, porém geram uma curiosidade no jogador e despertam a sensação de que alguém fazia algo próximo da agua antes de sua chegada.

4.10. Luz e sombras.

A luz e sombra são recursos indispensáveis no desenvolvimento, pois darão realidade e beleza única ao mundo. Apesar dessas vantagens, o *level designer* deve ter cuidado, pois erros na utilização de iluminação e sombras podem danificar a integridade do trabalho.

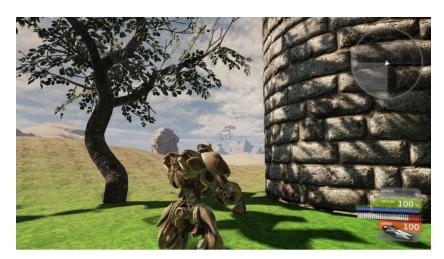


Figura 32 - Imagem demonstrando Luz e Sombra (do autor, 2013).

A luz do *level* desenvolvido foi feita com uma forte intensidade devido ao ambiente árido retratado. São notadas claramente na Figura 32, sombras fortes e acentuadas em arvores e pedras para retratarem desertos por serem lugares abertos e de altas temperaturas. Houve um cuidado para que as sombras do *level* se mantivessem apenas para uma direção evitando quebra de imersão do jogador e seguindo a irradiação do sol.

No trabalho desenvolvido foram utilizados todos os passos previamente apresentados no capitulo dois com base nas ideias de Galuzin (2011). Todo o *level* projetado e desenvolvido foi produzido ná ferramenta UDK com seus próprios elementos.

5. CONSIDERAÇÕES FINAIS

O objetivo desse trabalho é deixar claro a importância do *level design* na produção de um cenário. Assim como foi aqui apresentado, existem vários detalhes a se atentar, pois a gama de aspectos que compõem um nível leva a definir novos conceitos sobre a simplicidade da arte ao se criar cenários para jogos. Alguns desses conceitos foram aqui apresentados e demonstrados com a ferramenta UDK. Entende-se que o *level design* é um tema com pouca abordagem atualmente mas, ao utilizar os passos para se criar um cenário, o resultado será um trabalho facilitado e que permite a criação de ambientes esteticamente agradáveis e de melhor jogabilidade.

Pode-se dizer que atualmente é mais difícil criar um jogo sem um *level design*. Mesmo que esse seja um conceito novo, ele pode ser notado desde o início da produção de jogos. O que define que planejamento do *level design* não é sempre visível. Ele pode ser criado mesmo que o profissional responsável pelo desenvolvimento não tenha conhecimento do conceito para reproduzir esse trabalho mas isso não o torna dispensável. O mal desenvolvimento pode transformar uma grande ideia em uma frustação, onde o jogo pode vir a ter ótima historia e boa jogabilidade porém o atrativo visual deixa muito a desejar, fazendo os jogadores perderem o interesse.

O tema proposto é aplicável atualmente dada a necessidade de criar *levels* bem planejados e que acompanhem a velocidade do mercado atual. Para que esse trabalho seja facilitado, no desenvolvimento pratico aqui proposto, foi escolhida a engine UDK, pois com ela é possível criar *levels* agradáveis e com uma beleza atrativa de forma rápida e coesa.

Conclui-se que, para obter bons resultados em *level design*, é requerida a utilização de uma ferramenta facilitadora que auxilie durante produção. Ambos são de vital importância para os desenvolvedores de jogos que desejam atingir o mercado de forma expressiva, pois para se criar grandes produções ou até mesmo

jogos casuais é necessário um grande planejamento não somente do *level design* mais do GDD como um todo.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Codigo em UnrealScript. Disponível em:

http://udn.epicgames.com/Three/CameraTechnicalGuide.html. Acesso em: 11/05/2013e.

Content Browser. Disponível em:

http://udn.epicgames.com/Three/ContentBrowserReference.html. Acesso em: 12/05/2013h.

Dicionario Online Dicio Disponível em:

< http://www.dicio.com.br/ludico/ >. Acesso em: 03/06/2013

EPIC GAMES. Getting Started: Developing Mobile Projects. Disponível em:

http://udn.epicgames.com/Three/GettingStartedMobile.html. Acesso em: 11/05/2013c.

EPIC GAMES. Getting Started: Gameplay Elements. Disponível em:

http://udn.epicgames.com/Three/GettingStartedGameplay.html>. Acesso em: 12/05/2013g.

EPIC GAMES. Getting Started: Programming. Disponível em:

http://udn.epicgames.com/Three/GettingStartedProgramming.html>. Acesso em: 11/05/2013d.

EPIC GAMES. Getting Started: Unreal Engine 3. Disponível em:

http://udn.epicgames.com/Three/GettingStartedOverview.html. Acesso em: 11/05/2013a.

FEIL, Scattergood. Beginning Game Level Design. Editora: Course Tech, 2005.

GALUZIN, Alex. UltimateLevelDesignGuide. Editora: World of Level Design™.com, 2011.

HUIZINGA, Johan. Homo Ludens: O Jogo Como Elemento Da Cultura. São Paulo. Editora Perspectiva, 7ª edição, 2012.

Imagem do emulador de iOS da UDK. Disponível em:

http://udn.epicgames.com/Three/GettingStartediOSDevelopment.html>. Acesso em: 11/05/2013b.

Imagem do jogo Battlefield 3. Disponível em:

http://sticktwiddlers.com/2012/11/20/battlefield-3-contra-easter-egg-comes-to-light/. Acesso em: 10/04/2013.

Imagem do jogo Call of Duty®: Modern Warfare 3. Disponível em: http://www.callofduty.com/mw3/images/mw3-image-2011-020. Acesso em: 09/04/2013.

Imagem do jogo Deus EX. Disponível em: < http://www.deusex.com/media/images#. Acesso em: 27/04/2013.

Imagem do jogo Fable 2. Disponível em: http://lionhead.com/fable-2/>. Acesso em: 01/05/2013.

Imagem do jogo Grand Theft Auto. Disponível em:

http://www.rockstargames.com/V/screenshots/screenshot/731>. Acesso em: 01/05/2013.

Imagem do jogo Left for Dead 2. Disponível em:

http://techreport.com/review/17939/asus-ul80vt-14-inch-notebook/9>. Acesso em: 01/05/2013.

Imagem do jogo Little Big Planet Karting. Disponível em:

http://www.littlebigplanet.com/pt/media/category/littlebigplanet-karting. Acesso em: 09/04/2013.

Imagem do jogo Minecraft. Disponível em:

http://empireminecraft.com/threads/happy-valentines-day.21591/>. Acesso em: 01/05/2013.

Imagem do jogo The Witcher Wild Hunt. Disponível em:

http://www.thewitcher.com/>. Acesso em: 01/05/2013.

Imagem do jogo Tomb Raider. Disponível em: < http://www.tombraider.com/>. Acesso em: 01/05/2013.

Imagem do jogo Trine 2. Disponível em:

http://trine2.com/site/index.php?page=media>. Acesso em: 27/04/2013.

Interface da UDK. Disponível em:

http://udn.epicgames.com/Three/UnrealEdUserGuide.html. Acesso em: 18/05/2013j.

LEWIS, Michael; JACOBSON, Jeffrey. Games Engines: In Scientific Research. Editora: Communications of the ACM, Vol. 45, No. 1, 2002.

MCGUIRE, Morgan; JENKINS, Odest Chadwicke. Creating Games: Mechanics, Content, and Technology. Wellesley, Massachusetts: A K Peters, 2009.

ORRICO, Alexandre. Mercado brasileiro de games já é o quarto maior do mundo e deve continuar a crescer. Disponivel em < http://www1.folha.uol.com.br/tec/1165034-mercado-brasileiro-de-games-ja-e-o-quarto-maior-do-mundo-e-deve-continuar-a-crescer.shtml >. Acesso em: 03/03/2013.

PAULIN, Rafael. Level Design – Overview. Disponível em < http://www.pontov.com.br/site/index.php/game-design/73-leveldesign/286-level-design-overview>. Acesso em: 03/03/2013.

PLANTIER, Renato Duarte. Deserto: Características. Disponível em:

http://meioambiente.culturamix.com/recursos-naturais/deserto-caracteristicas>.

Acesso em: 14/05/2013.

Representação de programação feita no Kismet. Disponível em:

http://udn.epicgames.com/Three/KismetUserGuide.html. Acesso em: 11/05/2013f.

Unreal Frontend. Disponível em:

http://udn.epicgames.com/Three/UnrealFrontend.html>. Acesso em: 12/05/2013i.